

A Feedback-Based Implementation Scheme for Batch Process Optimization

E. Visser, B. Srinivasan, S. Palanki*, D. Bonvin ^{‡‡}

*Institut d'automatique, École Polytechnique Fédérale de Lausanne,
CH-1015 Lausanne, Switzerland.*

**Florida State University, Tallahassee, Florida, USA.*

The terminal-cost optimization of a control - affine nonlinear system leads to a discontinuous solution that can be characterized in a piecewise manner. To implement such an optimal trajectory despite disturbances and parametric uncertainty, a *cascade optimization scheme* is proposed in this paper, where optimal reference signals are tracked. Optimality is achieved by the appropriate definition of reference signals (input bounds, state constraints, or switching functions) to track in various sub-intervals. Furthermore, conservatism is introduced into the optimization problem to ensure satisfaction of path constraints in the presence of uncertainty. Finally, the proposed cascade optimization scheme is illustrated on a simulation of a fed-batch penicillin fermentation plant.

Key words: On-line optimization, Uncertainty, Feedback control, Optimal control, Batch process optimization

1 Introduction

A wide variety of specialty chemicals are made in batch reactors. To cope with competition, it is important to operate them in an optimal manner. The optimal operating policy for a given batch process is usually calculated under the assumption of a perfect model. However, realistic applications are subject to uncertainty in initial conditions, model mismatch, and process disturbances, all of which affect the optimal solution. This provides the motivation for *on-line* calculation and implementation of the optimal operating policy. Also, recent developments in sensor technology has opened up new directions in process control and optimization. Thus, the availability of frequent measurements suggests a paradigm shift from *open-loop model-based* optimization to *closed-loop measurement-based* optimization.

The two approaches to measurement-based optimization that have been proposed in the literature are discussed briefly below.

^{‡‡} Author to whom all correspondence should be addressed

Repeated optimization: Using this approach, the optimal inputs are updated by solving a finite horizon optimization problem at each time step. The parameters of the model required for the optimization are also estimated on-line. The solution to the optimization problem is either computed numerically [5] or analytically [8]. Some of the drawbacks of this approach include: (i) high computational burden, especially in the presence of state and input constraints; (ii) necessity of full-state information; (iii) possible infeasibility of the solution obtained [12]; (iv) conflict between parameter estimation which requires persistency of excitation and the optimization [11]; and (v) possible chattering of the feedback optimal solution [9].

Cascade optimization: The optimal set-point trajectory that corresponds to performance is computed. A ‘low level’ tracking controller ensures that the system does not stray very far away from the optimal trajectory [13,7]. In addition, a ‘high level’ optimizer is invoked periodically to ensure optimality despite disturbances. This scheme, in general, requires neither full state information nor on-line parameter estimation.

The cascade optimization approach combines the positive features of optimal operation and feedback control. The basis of the cascade optimization scheme is tracking. Since there are typically more states than inputs, one cannot guarantee that all the state trajectories will be accurately tracked in the presence of disturbances. In general, a single combination of states that can be tracked during the entire time interval does not exist. Instead, a piecewise definition of the outputs to track, i.e., different combinations of states/inputs for different time intervals, will be proposed.

Furthermore, conservatism has to be introduced in the optimization problem to guarantee feasibility in the presence of uncertainty. The use of margins from constraints, also called back-offs, and their computation as a function of the uncertainty will be developed within the framework of cascade optimization. The paper is organized as follows. Section 2 formulates the problem. Section 3 explains the cascade optimization framework in detail. The selection of outputs to track is treated in Section 4, and Section 5 discusses the conservatism required in the presence of uncertainty. Section 6 illustrates the approach via a simulated example, and Section 7 concludes the paper.

2 Problem Formulation

The end-point optimization of a nonlinear, control - affine batch process can be mathematically formulated as:

$$\min_{u(t)} J = \phi(x(t_f)) \quad (1)$$

$$s.t. \dot{x} = f(x) + \sum_{i=1}^m g_i(x) u_i, \quad x(0) = x_0 \quad (2)$$

$$S(x, u) \leq 0 \quad (3)$$

where u is the m -vector of manipulated inputs, x is the n -vector of states, $f(x)$ and $g_i(x)$ are n -dimensional analytic vector fields. Also, the vector fields g_i are assumed to be of full rank for all x . t_f is the final time, ϕ is a smooth scalar function and $S(x, u)$ is a σ -dimensional vector of path constraints. Note that problem formulation (1) - (3) does not include terminal constraints.

Application of Pontryagin's Minimum Principle (PMP) [3] to this end-point optimization problem involving a control-affine system, results in the following Hamiltonian:

$$H(x, u, \lambda) = \lambda^T \left(f(x) + \sum_{i=1}^m g_i(x) u_i \right) + \mu^T S(x, u) \quad (4)$$

$$\dot{\lambda}^T = -\frac{\partial H}{\partial x} = -\lambda^T \left(\frac{\partial f}{\partial x} + \sum_{i=1}^m \frac{\partial g_i}{\partial x} u_i \right) - \mu^T \frac{\partial S}{\partial x} \quad (5)$$

$$\lambda^T(t_f) = \frac{\partial \phi}{\partial x} \Big|_{t_f} \quad (6)$$

where $\lambda(t) \neq 0$ is the vector of costates and $\mu(t) \geq 0$ are the Lagrange multipliers for the state constraints and input bounds ($\mu_j > 0$ only when $S_j(x, u) = 0$, $j \in \{1, \dots, \sigma\}$).

The first-order necessary conditions for optimality are:

$$H_{u_i} = \lambda^T g_i(x) + \mu^T \frac{\partial S}{\partial u_i} = 0, \quad i = 1, \dots, m \quad (7)$$

H_{u_i} is independent of the inputs, if $S(x, u)$ is affine in the manipulated input vector, u . Thus, the necessary conditions, by themselves, cannot determine the inputs. For such problems, the optimal solution has the following properties:

- The inputs are in general discontinuous, yet inputs are analytic between discontinuities.
- The solution between two discontinuities will be referred to as an *arc*. Three types of arcs are possible:
 - (i) inputs determined by active input bounds;
 - (ii) inputs determined by active state constraints;

(iii) singular arc, when inputs are not determined by any of the active constraints.

- Analytic expressions for these arcs can be obtained, though the sequence and the switching times have to be computed numerically in most cases.

A piecewise analytic characterization of the optimal inputs helps to both improve the computational efficiency and to choose the implementation strategy. Whether or not an arc is singular depends on the function $\psi_i = \lambda^T g_i(x)$, which is referred to in the literature as the *switching function*. This function vanishes over the singular time interval. Outside the singular interval, the manipulated input $u_i(t)$ is on a state constraint or an input bound.

Methods for calculating the singular arcs are available in the optimal control literature [8]. Since the switching function is zero over a time interval, its derivatives with respect to time are also zero. Thus, a sequence of time differentiations is performed until the inputs $u_i(t)$ appear explicitly. The resulting expression is then solved for $u_i(t)$ in terms of x and λ .

3 Cascade Optimization Framework

A *cascade optimization* structure is proposed to incorporate feedback into the optimization framework (Figure 1). The ‘high level’ optimizer solves the optimization problem and selects the appropriate outputs to track for specific time intervals. Thus, it provides: (i) the feedforward inputs, u^* ; (ii) the reference signals, y^* ; and (iii) the switching strategy between various subsequent outputs. The optimizer constitutes the outer loop and is indicated by the thin lines in Figure 1. The reference signal of the corresponding output is then tracked with the help of the ‘low level’ feedback controller (inner loop - thick lines in Figure 1).

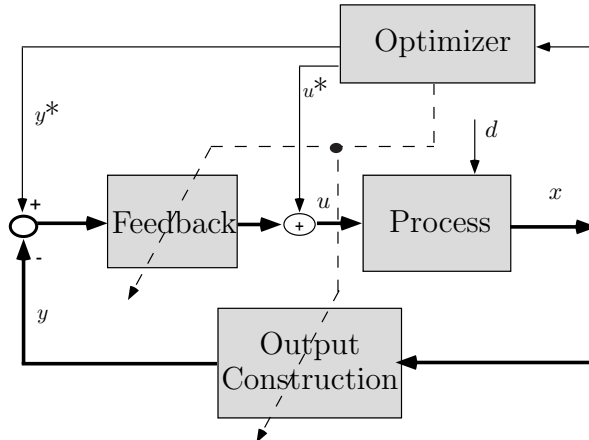


Fig. 1. Cascade Optimization

Due to the presence of the feedforward term u^* , the feedback is inactive in the absence of uncertainties (model mismatch, disturbances). However, in the presence of small uncertainties, the feedback ensures that the system does not stray far away from the optimal trajectory. To ensure optimality despite large

uncertainties, the reference signal and the switching strategy of the optimizer can be updated during the course of a run. In addition, if runs are repeated, the optimizer can adapt itself on a run-to-run basis.

Though the *cascade optimization* framework is quite general, most of the issues that follow will be restricted to terminal-cost optimization of control - affine nonlinear systems. The restriction is motivated by the fact that powerful geometric control concepts can be utilized for such systems.

The two goals of the optimizer are to: (i) select appropriate outputs to track; and (ii) calculate the reference signals numerically so as to guarantee feasibility. The main issues discussed next concerns these two goals of the optimizer. When the solution lies on the feasibility boundaries, there is no maneuverability to implement the feedback proposed in the cascade optimization framework. In particular, care should be taken to ensure that the constraints are not violated. This calls for the introduction of conservatism.

4 Selection of Outputs to Track

4.1 Definition of outputs

The ‘high level’ optimizer provides an open-loop solution which consists of input and state trajectories that minimize the objective function at the final time. In the absence of uncertainty, the tracking of *any* state will result in all the other states evolving on their optimal trajectories. However, in the presence of uncertainty, this is no longer true. In fact, since there are m inputs, only m states or combination of states (outputs) can be kept at desired values over time.

Secondly, since the disturbances take the system to a state different from that expected in the nominal condition, it is interesting to know the optimal trajectory from that new operating point onwards. Is tracking still a viable option to achieve optimality ? If so, what are the outputs or the combination of states that need to be tracked ? The answer to this question is provided in the following proposition.

Theorem 1 *Consider the end-point optimization problem (1) - (3) for which measurements or estimates of all the states and costates are available. Feedback optimality is achieved by:*

- (i) *open-loop application of the input when an input bound is active;*
 - (ii) *ideal tracking of the state constraint when a state constraint is active;*
 - (iii) *ideal tracking of switching functions during a singular interval;*
- along with appropriate switching between the various sub-intervals.*

Proof: The problem of feedback optimality is equivalent to that of satisfying the necessary conditions, which in turn, is reformulated as the problem of tracking $H_u = 0$. Tracking $H_u = 0$ has different interpretations with respect to the three types of arcs and is discussed below.

- In the case where an input is determined by its bound, assuming, without loss of generality, that the bound corresponding to μ_1 is active, i.e., $H_{u_i} = \psi_i + \mu_1 = 0$. As long as $\psi_i < 0$, $H_{u_i} = 0$ implies that $\mu_1 = -\psi_i$. The optimal solution remains on the active input bound as long as μ_1 is non-zero.
- If the input is determined by a state constraint, following the same argument, it is optimal to keep the constraint active as long as the ψ_i does not change sign. In the input bound case, it is straightforward to determine the input that keeps the constraint active. In contrast, an algorithmic approach is required to keep a state constraint active. Hence, optimality is achieved by choosing the output $y = S(x, u)$ with a zero set point.
- For a singular arc, Condition (7) reduces to $H_{u_i} = \psi_i = 0$. Therefore, the choice of $y = \psi_i$ with a zero set point ensures optimality.

QED

An important point to note is that there is no single combination of states that can be tracked during the entire optimization interval. This is due to the fact that tracking $H_u = 0$ means tracking different combinations of states in different types of intervals.

The result provided in Proposition 1 not only chooses the outputs to track but also performs the input - output pairing. An active state constraint or the switching function is differentiated with respect to time until an input appears explicitly. The first input to appear is then paired with the appropriate output.

The numerical computation of the optimal solution has two parts, i.e., (i) the switching instants, and (ii) the value of the inputs between the switching instants. The proposition presented above deals only with the second problem, while the first calls for a periodic reoptimization.

4.2 Controllability and robustness issues

Having selected the outputs to track, the first issue that needs to be addressed is whether or not the outputs can be tracked. It is shown below that the outputs are controllable and are robust with respect to parametric uncertainty.

4.2.1 Controllability

The outputs chosen for tracking should be controllable. A recent result [14] states that input - affine nonlinear systems *always* lose first-order differential controllability along the optimal solution. This implies that a mode exists in the optimal solution that cannot be controlled to its desired trajectory. However, it can be shown that the outputs chosen by Proposition 1 are controllable. This can be shown as follows: (i) during open-loop application, the issue of

controllability does not arise; (ii) when on a state constraint ($y = S(x, u)$), or along a singular arc ($y = \psi_i$), the inputs can in general be obtained by a finite number of repeated differentiations of y . In such a case, y is controllable because of this explicit relationship between u and y . If the inputs do not appear when deriving y an infinite number of times, the inputs are non-unique and the output tracking problem is ill-posed. For details, see [2].

4.2.2 Robustness

The cost sensitivity to non-optimal operation is in general much lower along a singular arc than along constrained arcs. Consider the optimal solution determined by a constraint where $H_{u_i} = \lambda^T g_i(x) + \mu^T \frac{\partial S}{\partial u_i} = 0$. When the input deviates from this constraint, the constraint is no longer active and the corresponding μ_i becomes zero. Thus, the change in cost is directly proportional to $\lambda^T g_i(x) \neq 0$. In contrast, along a singular arc, $H_{u_i} = \lambda^T g_i(x) = 0$. Since the first order gradient H_{u_i} is zero, any deviation of u_i from the optimal trajectory will cause a relatively small loss in cost.

Alternatively, the optimal output for a path constrained region is independent of the system parameters and requires only a reduced number of state estimates for its implementation. In contrast, along a singular arc, the dependence on model parameters might be high and require full-state knowledge. However, the effect of errors in the singular region on the overall cost is negligible.

Table 1 summarizes the sensitivity discussion. It can be concluded that the cascade optimization scheme chooses outputs which are not sensitive to parameters in regions where the cost is most sensitive. Thus, the cascade optimization scheme is quite robust to parametric variations.

Sensitivity	Input determined by		Singular
	Input Bound	State Constraint	Input
of Cost to Suboptimal Input	High	High	Low
of Output to Uncertainty	No	No	Yes

Table 1
Robustness issues

4.3 Optimal output in the singular region

First the on-line construction of the switching function through the estimation of costates will be developed. Then, the relevance of the availability of good measurements is discussed.

4.3.1 Estimation of costates

The major problem in the approach proposed is the estimation of the costates for tracking the switching function. The best scenario is when analytical expressions for the costates in terms of the present states are available, as in the

‘sweep method’ ($\lambda = \mathcal{P}x$) used in the linear quadratic regulator. This is not the case for the class of problems considered here.

Here, it is assumed that all the states are available. If full state measurement is not available, an appropriate state estimator needs to be set up. Consider the case where all inputs are singular during the entire interval. An estimate $\hat{\lambda}(t)$ of $\lambda(t)$ is obtained with the observer

$$\dot{\hat{\lambda}}(t) = -A^T(x)\hat{\lambda}(t) - G(\hat{\lambda}(t_f|t) - \bar{\lambda}(t_f|t)) \quad (8)$$

$$A(x) = \frac{\partial f}{\partial x}(x) + \sum_{i=1}^m \frac{\partial g_i}{\partial x}(x)u_i^* \quad (9)$$

$$\hat{\lambda}(t_f|t) = \mathcal{T}_{t,t_f}^{-1}(x)\hat{\lambda}(t), \quad \bar{\lambda}(t_f|t) = \left. \frac{\partial \phi}{\partial x} \right|_{\hat{x}(t_f|t)} \quad (10)$$

$$\mathcal{T}_{t,t_f}(x) = e^{\int_t^{t_f} A(x(\tau)) d\tau}, \quad \hat{x}(t_f|t) = x^*(t_f) + \mathcal{T}_{t,t_f}(x(t) - x(t)^*) \quad (11)$$

where x^* and u_i^* are the nominal state and input solutions to Problem (1) - (3). The estimated states $\hat{x}(t_f|t)$ and costates $\hat{\lambda}(t_f|t)$ at terminal time are predicted using a state transition matrix, \mathcal{T}_{t,t_f} , to avoid having to repeat explicit integration of the state and costate dynamic equations until final time at each time step. Note that the costate transition matrix is the inverse of the state transition matrix. From the terminal states $\hat{x}(t_f|t)$, the costates required for the sake of optimality, $\bar{\lambda}(t_f|t)$, are also calculated.

The gain matrix G is used to control the speed of correction. This is important since, if $G = 0_{n \times n}$, the λ trajectory deviates from its nominal value and there is no guarantee that the final conditions on λ are met. Note that it is more important to satisfy the final conditions on λ than the corresponding initial conditions. Thus, a closed-loop observer is proposed to force the costates at final time to their desired values.

The next complexity lies in the determination of $\mathcal{T}_{t,t_f}(x)$. As a first approximation, the nominal trajectory can be used to calculate the state transition matrix, i.e., $\mathcal{T}_{t,t_f}(x) \approx \mathcal{T}_{t,t_f}(x^*)$. In some cases, this is insufficient, and a first-order Taylor series expansion can be used $\mathcal{T}_{t,t_f}(x) \approx \mathcal{T}_{t,t_f}(x^*) + (x - x^*) \frac{\partial \mathcal{T}}{\partial x}$.

When singular arcs are concatenated with other types of arcs, then prediction of the costates at final time becomes more involved. In such a case, the costates at the end of the singular interval can be used instead of final time. The issue gets even more involved in the presence of state constraints, since the costates can be discontinuous when entering or leaving the constraints.

4.3.2 Measurements

The costate estimation relies in general on full-state measurement which is not always available. Also, some measurements may only be retrievable from the process with considerable delays. In such a case, the missing data may be estimated with a suitable state estimator. However, as discussed above, the cost is rather insensitive to errors resulting from model or measurement

uncertainty along singular arcs. In the example section below, it will moreover be shown how the costate estimation scheme can be replaced by simple ad-hoc adjustments of the singular input. Accurate full-state measurement is thus not essential in the singular case.

On the contrary, when process operation is limited by state constraints, it is essential to have accurate measurements or estimates of the states involved in the constraint expressions as the process has to be operated as close as possible to the constraints for optimal performance. Special instrumentation or algorithmic estimation efforts have to be provided in case of insufficient knowledge of this in general reduced set of states.

5 Conservatism to handle uncertainty

Here the feasibility of constraints in the presence of parametric uncertainty and disturbance is considered next. The problem formulation is as follows:

$$\begin{aligned} \min_{u(t)} \quad & \Phi(x(t_f)) \\ \text{s.t.} \quad & \dot{x} = f(x, \theta) + \sum_{i=1}^m g_i(x, \theta) u_i + d, \quad x(0) = x_0 \\ & S(x, u) \leq 0 \end{aligned} \tag{12}$$

where d denotes a vector of randomly distributed process noise with zero mean and covariance v_d . Assume that the disturbance is not correlated over time, i.e. $E(d(\tau) d^T(\tau')) = \delta(\tau - \tau') \text{diag}(v_d)$, where δ denotes the Dirac function. θ represents the set of uncertain parameters. The uncertainty description can be either probabilistic or of the set membership type. In the former, θ is associated with the probability distribution function $p(\theta)$, whilst in the latter, the only information available is $\theta \in \Theta$, where Θ is a bounded set. Note that the constraint expression does not explicitly depend on Θ : $S(x, u) \neq S(x, u, \theta)$.

5.1 Back-off calculations

The solution to the optimization problem (12) depends on the realization of the disturbance d and the value of θ , both of which are unknown. Thus, to ensure feasibility of the path constraints despite the variations in d , back-offs from the constraints, which depend on the noise variance v_d , have to be introduced. To calculate the back-offs amidst parametric uncertainty, two design values $\bar{\theta}$ and v_θ are necessary. They represent the mean and variance in the case of probabilistic uncertainty description, $\bar{\theta} = E[\theta]$, $v_\theta = E[(\theta - E[\theta])^2]$. When the uncertainty is of the set membership type, the worst case scenario is considered, $\bar{\theta} = \arg \max_{\theta} S(x(\theta), u)$, $v_\theta = 0$.

The optimization then becomes:

$$\begin{aligned} \Phi_b &= \min_{u(t)} \Phi(x(t_f)) \\ \text{s.t. } \dot{x} &= f(x, \bar{\theta}) + \sum_{i=1}^m g_i(x, \bar{\theta}) u_i, \quad x(0) = x_0 \\ S(x, u) + b_S &\leq 0 \end{aligned} \tag{13}$$

The margins, or back-offs, b_S are related to the uncertainty (disturbance and parametric errors) and will be calculated below. Note that the nominal solution (x^*, u^*) to Problem (13) is obtained with the uncertainty being eliminated from the system dynamics. The state evolution around the nominal trajectory x^* under the effect of uncertainty is considered *via* the back-offs in the path constraints. Since it has been assumed that d is zero-mean gaussian, Φ_b represents, to a first-order approximation, the cost averaged over all the realizations of d .

The back-off expressions will now be derived. Let $x(t)$ and $u(t)$ denote the actual states and inputs of the system and $\Delta x(t) = x(t) - x^*(t)$, $\Delta u(t) = u(t) - u^*(t)$, $\Delta \theta = \theta - \bar{\theta}$. The evolution of $\Delta x(t)$ can be calculated by the linear time-varying state space model:

$$\dot{\Delta x}(t) = A(t) \Delta x(t) + B(t) \Delta u(t) + P(t) \Delta \theta + d(t) \tag{14}$$

$$A(t) = \left(\frac{\partial f}{\partial x} + \sum_{i=1}^m \frac{\partial g_i}{\partial x} u_i^* \right) \Big|_{x^*} \tag{15}$$

$$B(t) = [g_1 \ g_2 \ \cdots \ g_m] \Big|_{x^*} \quad P(t) = \left(\frac{\partial f}{\partial \theta} + \sum_{i=1}^m \frac{\partial g_i}{\partial \theta} u_i^* \right) \Big|_{x^*} \tag{16}$$

Let the feedback provided by the cascade optimization structure be $u = u^* + u_{fb}(y, y^*, \kappa)$, where κ are the feedback controller parameters. The feedback is analyzed in a neighbourhood of the nominal trajectory using the linear approximation: $\Delta u(t) = -K(t) \Delta x(t)$, where $K(t) = -\frac{\partial u_{fb}}{\partial x}$. The variance of $\Delta x(t)$, $v_x = E[\Delta x(t) \Delta x(t)^T]$, can be calculated as in [4] and reads:

$$v_x(t) = v_d \int_0^t \Phi(\tau) \Phi^T(\tau) d\tau + v_\theta \int_0^t \Phi(\tau) P(\tau) P^T(\tau) \Phi(\tau) d\tau \tag{17}$$

where $\Phi(\tau) = e^{\int_\tau^t [A(t') - B(t') K(t')] dt'}$. The linearization of the constraints gives:

$$S(x, u) = S(x^*(t), u^*(t)) + \left(\frac{\partial S}{\partial x} - \frac{\partial S}{\partial u} K(t) \right)^T \Delta x(t)$$

The back-off size is calculated such that feasible operation is ensured with the probability α . For this, a factor $\beta = \sqrt{2} \text{erf}^{-1}(2\alpha - 1)$ is introduced, where erf^{-1} is the inverse error function of the normal distribution:

$$b_S = \beta \left(\left| \frac{\partial S}{\partial x} \right| + \left| \frac{\partial S}{\partial u} \right| |K(t)| \right)^T \sqrt{\text{diag}(v_x(t))} \quad (18)$$

In practical applications, output measurements are also corrupted by noise: $y = h(x, u) + \eta$. Let the noise covariance matrix be v_η . This can be taken into account in the conservative design by replacing $\sqrt{\text{diag}(v_x)}$ by $\sqrt{\text{diag}(v_x) + \text{diag}(v_\eta)}$ in (18).

The back-offs needed in the open-loop case can be computed with (18) and setting $K = 0_{m \times n}$. Note that if the system matrix $A(t)$ is unstable, the open-loop variance of x may be large, requiring a lot of conservatism. The introduction of a feedback gain will stabilize the linearized system, thereby reducing v_x . Alternatively, introduction of feedback increases $\left| \frac{\partial S}{\partial u} \right| |K(t)|$, which is zero in the open-loop case. Feedback also introduces coupling between different inputs. In the process of reducing the back-off for S_i , b_{S_i} , the back-off for S_j , b_{S_j} , might be increased. It is even possible that an inactive constraint S_j becomes active due to an increase of b_{S_j} . Thus, a compromise between $u^*(t)$ and $K(t)$ has to be sought in the solution to Problem (13).

5.2 Feedback design

The nominal input profile $u^*(t)$ and feedback controller parameters κ are found by solving the following optimization problem. Note that the back-offs b_S given by (18) implicitly depend on κ :

$$\begin{aligned} \Phi_b^{cl} &= \min_{u(t), \kappa} \Phi(x(t_f)) \\ \text{s.t. } \dot{x} &= f(x, \bar{\theta}) + \sum_{i=1}^m g_i(x, \bar{\theta}) u_i, \quad x(0) = x_0 \\ S(x, u) + b_S &\leq 0 \end{aligned} \quad (19)$$

The following proposition is now formulated:

Theorem 2 *Let Problem (13) be feasible and let there exists a κ such that the feedback $u_{fb}(y, y^*, \kappa) = 0, \forall y$. Then the optimal cost Φ_b^{cl} of Problem (19) is less than or equal to the optimal cost Φ_b of Problem (13).*

Proof :

By choosing the κ which corresponds to $u_{fb}(y, y^*, \kappa) = 0, \forall y$, the feedback is removed and Problems (13) and (19) are equivalent. Since it is assumed that optimization problem (13) is feasible, (19) also has a feasible solution. Due to this inclusion, and since (19) has an additional degree of freedom, $\Phi_b^{cl} \leq \Phi_b$.

QED

It has been observed that the sensitivity of the cost function with respect to the parameters κ and the input vector $u(t)$ are in general different. This leads to conditioning problems in the numerical optimization. Therefore, instead of solving the optimization problem (19) which takes $u(t)$ and κ simultaneously

as decision variables, it is preferable to solve it in an optimization structure where κ and $u(t)$ are iterated in an inner-outer optimization structure:

$$\begin{aligned}
\min_{\kappa(t)} \Phi(x(t_f)) &= \Phi'(\kappa(t), \tilde{u}) \\
\text{s.t. } \tilde{u}(t) &= \arg \min_{u(t)} \Phi(x(t_f)) \\
\dot{x} &= f(x, \bar{\Theta}) + \sum_{i=1}^m g_i(x, \bar{\Theta}) u_i, \quad x(0) = x_0 \\
S(x, u) + b_S &\leq 0,
\end{aligned} \tag{20}$$

6 Case Study: Penicillin Fed-Batch Fermentation

In this section, the cascade optimization scheme is applied to the simulation of a penicillin fed-batch fermentation process and is compared with both open-loop implementation and on-line optimization. Other biotechnology examples have been treated in [14,15].

6.1 Problem description

The kinetics of penicillin fermentation has been studied in [1]. These authors proposed a simple model that is consistent with the observed functional dependencies of the specific growth rates, glucose uptake and penicillin formation, and estimated the model parameters to fit the experimental data. The model can be described as follows:

$$\dot{X} = \alpha(S, X) X - F \frac{X}{V} \tag{21}$$

$$\dot{S} = -\alpha(S, X) \frac{X}{Y_x} - \theta(S) \frac{X}{Y_p} - M_x X + F \frac{S_l - S}{V} \tag{22}$$

$$\dot{P} = \theta(S) X - K P - F \frac{P}{V} \tag{23}$$

$$\dot{V} = F \tag{24}$$

with F being the feed rate, S_l the feed concentration of the limiting substrate, X , S and P are the concentrations of biomass, substrate and penicillin respectively, and V the volume. A Contois law is used for the specific growth rate $\alpha(S, X) = \frac{\alpha_m S}{S + K_l X}$. Y_x and Y_p are constant yield coefficients and M_x represents the cell maintenance term. The formation of penicillin is described by a substrate inhibition model $\theta(S) = \frac{\theta_m}{1 + \frac{K_p}{S} + \frac{S}{K_i}}$ and K is the first-order penicillin

decay rate. The model parameters are listed in Table 3. The model (21)-(24) presents a nonlinear affine-in-input structure with $x = [X \ S \ P \ V]^T$ and $u = F$. The operational constraints $S \leq S_m$ and $X \leq X_m$ are considered to avoid induction of unwanted side reactions and for oxygen limitation respectively.

It is desirable to maximize the penicillin concentration for a given final time. The optimization problem reads thus:

$$\begin{aligned}
& \min_{u(t)} -P(t_f) \\
& \text{s.t. } \dot{x} = f + gu, \quad x(0) = x_0 \\
& \quad S - S_m \leq 0, \quad X - X_m \leq 0 \\
& \quad 0 \leq u \leq u_m
\end{aligned} \tag{25}$$

The process parameters Y_x and S_l are considered uncertain within the range listed in Table 4. Thus, the parameter uncertainty is of the set membership type. The nominal parameters (Table 3) are chosen in such a way that the substrate and biomass constraints will be satisfied for all possible combinations of Y_x and S_l when the nominal input is implemented on the process: $\bar{\theta} = \arg \max_{\theta} S(x(\theta), u)$, $v_{\theta} = 0$.

The process is also perturbed by a random normally distributed disturbance d with standard variation $\sigma_d = [0 \ 0.02 \ 0 \ 0]^T$: $\dot{x} = f(x) + g(x)u + d$. The disturbance is physically motivated by additional fluctuations in the substrate inlet flow. Additionally, it is supposed that full-state measurements are available and, furthermore, are corrupted by normally distributed noise η with standard deviation $\sigma_{\eta} = [2 \times 10^{-1} \ 2 \times 10^{-3} \ 4 \times 10^{-2} \ 1]^T$. Note that full-state measurements are only required for tracking the switching function in the singular region.

6.2 Nominal solution

Using Pontryagin's Minimum Principle [3], it can be shown that the optimal solution in fact consists of 6 arcs:

- (i) an upper input bound arc for $t \in [0 \ t_1^s]$;
- (ii) a substrate constraint arc for $t \in [t_1^s \ t_2^s]$;
- (iii) a lower input bound arc for $t \in [t_2^s \ t_3^s]$;
- (iv) a biomass constraint arc for $t \in [t_3^s \ t_4^s]$;
- (v) a singular arc for $t \in [t_4^s \ t_5^s]$;
- (vi) a lower input bound arc for $t \in [t_5^s \ t_f]$.

The switching times are listed in Table 4. Some optimal trajectories and the switching function are illustrated in Figure 3.

The optimal solution first lies on the upper input bound until the substrate constraint $S = S_m$ is reached. Then, it stays there in order to maximize the biomass growth rate. Just before the biomass constraint is reached, the substrate level is lowered to the value S_e for the following reason. When the biomass constraint $X = X_m$ is entered, it has to be tracked with the input $u = \alpha(S, X) V$ (see equation (21) with $\dot{X} = 0$). When such an input is applied, it can be seen that the substrate dynamics (22) is unstable. To remain bounded, the biomass constraint has to be entered with the substrate value of S_e which corresponds to the equilibrium point of the internal dynamics. The

biomass constraint is exited again towards the end of the batch on a singular arc. A short arc with $u = 0$ terminates the batch since the final time cost is sensitive to dilution.

Following the argument, it can be seen that the switching times t_1^s to t_3^s can be determined analytically, whilst t_4^s and t_5^s have to be determined numerically. The maximum penicillin concentration obtained with the set of parameters listed in Table 3 is $8.23 \frac{g}{l}$.

6.3 Open-loop implementation

In the presence of uncertainty, the open-loop input profile has to be designed and implemented with a certain conservatism in order not to violate the constraints.

The back-offs are computed in the way exposed in Section 5 and using $v_\theta = 0$ since the parametric uncertainty is of set membership type. The substrate and biomass profiles with the necessary back-offs can be seen in Figure 4 for different parameter variations and disturbance realizations. Note the upper envelope for the substrate uncertain evolution as predicted with formula (18). For the case of the nominal parameters $Y_X = 0.47 \frac{1}{h}$ and $S_l = 400 \frac{g}{l}$, the back-off from the biomass constraint is very small due to fast and stable substrate dynamics during this batch phase.

It is observed that the substrate concentration is considerably lower than S_m which leads to a reduced growth rate. This causes a large loss in penicillin productivity during the biomass constrained phase.

6.4 On-line optimization

A fine parametrization of the input was required during the initial batch phases to model the sharp switching between constraints and to quickly correct the perturbed state profiles at the beginning of each reoptimization task. A piece-wise constant input parametrization with 80 elements was adopted for the time interval $t \in [0, 100]$ hours, whereas 20 elements were used on the remaining interval $t \in [100, 150]$ hours. The optimization problem was solved using a Sequential Quadratic Programming (SQP) routine in Matlab. The SQP routine is time-consuming and frequently converged to a local minimum; thus, it had to be restarted and manually guided to the optimal solution.

When the process is reoptimized, direct measurements of the process states are used to estimate the uncertain parameters on-line. It is supposed that the inlet substrate concentration S_l can be measured and the yield factor Y_X determined from $Y_x = \frac{XV - X_0V_0}{S_lV - S_0V_0}$, under the assumption that most of the substrate is indeed consumed by the biomass.

The back-offs can be reduced through on-line optimization since feedback is provided by the reoptimization algorithm. Formulas (17) and (18) elaborated in Section 5 can be suitably adapted for the back-off design along the substrate

constraint in this case. Indeed, at each reoptimization, the algorithm will attempt to bring the system back to the constraint if a deviation from $S = S_m$ occurred. This feedback can then be modeled as the correction provided by a proportional controller $\Delta u(t) = -K(t) \Delta x(t)$ which tracks the substrate constraint. For a detailed derivation, see [14].

Since the back-off due to the disturbance is negligible along the biomass constraint, no back-off from this constraint was used with this scheme. The performance of the on-line optimization scheme was found to be better the higher the reoptimization frequency (see Table 2).

6.5 Cascade optimization with costate tracking

The proposed cascade optimization scheme was implemented in the following manner:

- When the input is on a bound (arcs (i),(iii) and (vi)), no feedback is applied.
- A PI-controller tracks the substrate constraint in arc (ii). In arc (iv), a controller (Figure 2) involving two cascaded PI controllers corrects any possible offset from the biomass constraint and tracks the biomass at its upper bound. This structure was necessary to stabilize the substrate internal dynamics and cope with the large time-scale differences between the biomass and substrate dynamics. Small back-offs from the state constraints are used to cope with the noise effects. The sampling time is 5 minutes.

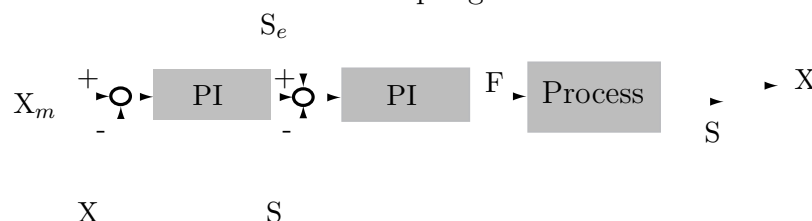


Fig. 2. Biomass controller: S_e is the off-line computed equilibrium value of the internal dynamics

- The costate estimation scheme requires the computation of the final costate. As the final nonsingular arc is very short, the end of the singular arc is considered as the final time. The switching function was tracked with a suitably tuned PI-controller. No on-line parameter estimation is carried out. The G matrix in (8) is chosen in such a way that the costate estimation scheme is stable and the estimation transient fast.
- The switching instants t_1^s to t_3^s are adjusted on-line since they can be computed analytically. However, those switching instants which could not be characterized analytically, i.e., the switching times t_4^s to t_5^s are left at their off-line computed values.

With the feedback design chosen, the back-offs for the cascade scheme were considerably reduced in comparison to the open-loop case so that the substrate

and biomass could be tracked close to their constraints. This can be seen in the simulations shown in Figure 5. The PI controller parameters for the substrate constraint tracking were optimized iteratively. It was not necessary to optimize the parameters for the cascade controller along the biomass constraint since the back-off given by the noise level was already very small. Note that the back-off is considerably reduced compared to the open-loop case (Figure 4). The switching function is tracked at its zero setpoint.

6.6 Cascade optimization without costate tracking

Note that in all batch phases apart from the singular arc, no parameter estimates and only partial state measurement are needed to achieve optimal performance. It was therefore investigated whether tracking the switching function could be replaced by a simpler control strategy requiring no full-state knowledge and parameter estimates.

From the off-line calculated optimal solution, it is seen that the input is continuous as it enters the singular arc. Thus, the off-line computed singular input along with an additional offset is applied. The offset is adjusted to guarantee continuity. This adaptation of the input is expected to produce a good approximation of the singular input which maintains the substrate level to values for which the penicillin growth rate is optimal.

6.7 Performance comparison

Table 2 illustrates the average performance of the various implementation methods for different parameter values. Note that the indicated costs correspond to average costs that are supposed to equal those obtained by solving Problem (25) with the corresponding back-offs for each implementation case.

Y_x	S_l	OL	On-line opt. with n_o				CO.co.	CO.no.	Ideal
			2	4	64	150			
0.43	360	7.14	7.55	7.87	7.91	7.94	7.94	7.91	7.95
0.47	360	7.43	7.75	7.91	7.94	7.99	7.99	7.98	8.00
0.45	380	7.67	7.91	7.95	8.07	8.09	8.09	8.09	8.10
0.43	400	7.89	8.04	8.14	8.17	8.18	8.19	8.18	8.20
0.47	400	8.19	8.19	8.19	8.21	8.22	8.22	8.22	8.23

Table 2

Performance comparison: OL = Open loop; n_o = number of optimizations; CO.co. = Cascade optimization with costate tracking; CO.no. = Cascade optimization without costate tracking; Ideal = Ideal solution with neither noise nor back-off

From any row in Table 2, the effect of back-offs on the average performance can be observed. The back-offs are largest for the open loop case and are

reduced progressively with increasing reoptimization frequency. The smallest back-offs are obtained within the cascade optimization scheme. Thus, the average performance augments from the open-loop implementation case to the on-line optimization scheme with increasing reoptimization frequency.

It is observed that a significant performance improvement is already obtained with a few reoptimizations. However, a total number of 150 optimizations, i.e. one reoptimization every hour, would be necessary to achieve the performance of the cascade optimization scheme.

The benefit of cascade optimization over open-loop implementation is considerable. The improvement in performance is primarily achieved through reduction of the offset from the biomass constraint.

When the cascade optimization strategy without costate tracking is considered, the performance is only slightly lower than with the complete cascade optimization scheme. Consequently, the difficulties linked to the costate estimation scheme can easily be avoided by a simple modification of the feedback strategy. Hence, the cascade optimization scheme is able to achieve near-optimal performance despite model errors by using only a few state measurements.

The best performance for each parameter set was obtained by recomputing the optimal solution with neither noise nor back-off. The performances of the cascade optimization scheme and on-line optimization with a total number of 150 optimizations are very good and differ only insignificantly from the optimal solution without back-off.

7 Conclusions

A cascade optimization scheme has been presented to implement an optimal solution in the presence of uncertainty. The scheme relies on tracking appropriate outputs during different time intervals. No feedback is applied when an input is determined by its bound. The outputs to track are determined from the state constraints when the optimal solution is determined by them. The switching function constitutes the optimal output when the solution is singular. Finally, feasibility of constraints is ensured despite uncertainty by introducing margins from constraints.

The feedback scheme was applied to a simulated penicillin fed-batch fermentation process subject to disturbance and parameter variations. The optimal solution was found and verified using Pontryagin's Minimum Principle. The performance of the cascade optimization scheme was significantly superior to that obtained by open-loop application of the off-line computed optimal solution. This was achieved by using only substrate and biomass measurements and standard PI controllers. No numerical reoptimization or parameter estimation was required, making the performance of this scheme robust to model and estimation errors.

Thus, the proposed cascade scheme constitutes a valuable improvement over

standard on-line optimization techniques by providing fast and robust feedback which can be implemented in a well-instrumented industrial environment. Future research will focus on how to extend application of the proposed scheme to optimization problems involving terminal constraints.

Acknowledgements

The authors acknowledge the financial support of the Swiss National Science Foundation through the Project 21-46922.96.

References

- [1] Bajpai, R.K., Reuss, R., Evaluation of Feeding Strategies in Carbon-Regulated Secondary Metabolite Production through Mathematical Modelling, *Biotechnology and Bioengineering*, Vol. 13, pp. 717-738, 1981.
- [2] Baumann, T., Infinite Order Singularity in Terminal Cost Optimal Control: Application to Robotic Manipulators, Thesis *N^o 1778*, Ecole Polytechnique Fédérale de Lausanne, Lausanne, 1998
- [3] Bryson, A.E., Ho, Y.- C., *Applied Optimal Control*, Revised Printing, Hemisphere Publishing Corporation, New York, USA, 1975
- [4] Chen, C.T., *Linear System Theory and Design*, Holt-Saunders, New York, 1970
- [5] Eaton, J.W., Rawlings, J.B., Feedback Control of Nonlinear Processes using On-Line Optimization Techniques, *Comp.Chem.Eng.*, Vol. 14, pp. 469-479, 1990
- [6] Kolavennu, N.S, Palanki, S., Cockburn, J.C., Nonlinear Controller Design by Uncovering Linear Dynamics Hidden by Input-Output Linearization, *Proceedings ADCHEM 1997*, pp. 511-516, Banff, 1997
- [7] Krothapally, M., Palanki, S., A Neural Network Strategy for Batch Process Optimization, *Comp.Chem.Eng.*, Vol. 21, pp. 463-468, 1997
- [8] Palanki, S., Kravaris, C., Wang, H.Y., Synthesis of State Feedback Laws for End-Point Optimization in Batch Processes, *Chem.Eng.Science*, Vol. 48, 1, pp. 135-152, 1993
- [9] Robbins, H.M., A Generalized Legendre-Clebsch Condition for the Singular Cases of Optimal Control, *IBM Journal*, pp. 361-372, July 1967
- [10] Sampath, V., Palanki, S., Cockburn, J.C., Robust Nonlinear Control of Polymethylmethacrylate Production in a Batch Reactor, *Comp.Chem.Eng*, Vol. 22, pp. S451-S457, 1998
- [11] Sastry, S., Bodson, M., *Adaptive Control: Stability, Convergence and Robustness*, Prentice-Hall, London, 1989
- [12] Rawlings, J.B., Tutorial: Model Predictive Control Technology, *Proceedings of the 1999 American Control Conference*, Vol. 1, pp. 662-76, San Diego, 1999

- [13] Srinivasan, B., Visser, E., Bonvin, D., Optimization-based Control with Imposed Feedback Structures, *Proceedings ADCHEM'97*, pp. 635-640, Banff, 1997
- [14] Visser, E., *A Feedback-based Implementation Scheme for Batch Process Optimization*, Thesis N^o 2097, Ecole Polytechnique Fédérale de Lausanne, Lausanne, 1999
- [15] Visser, E., Srinivasan, B., Palanki, S., Bonvin, D., On-line Optimization of Batch Processes by Tracking States and Costates, *Proceedings IFAC World Congress*, Vol. N, pp. 31-36, Beijing, 1999

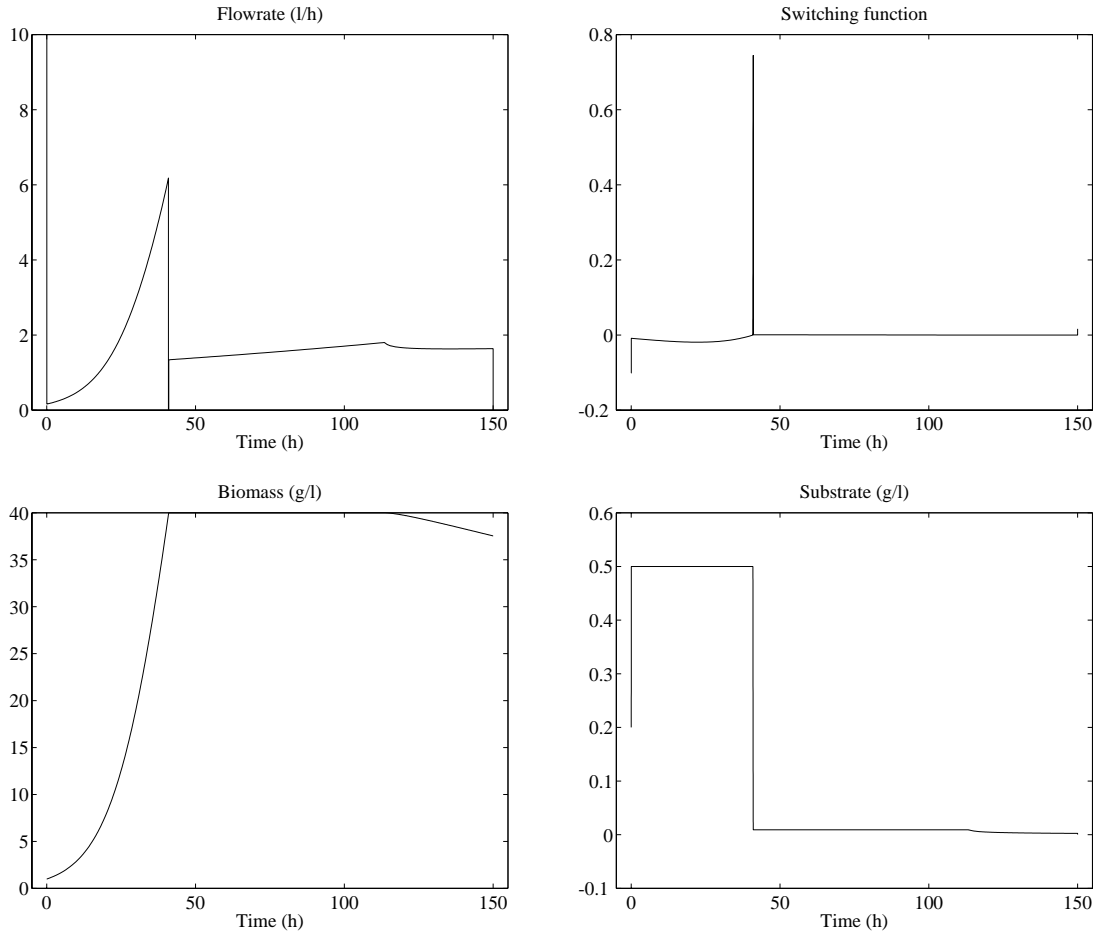


Fig. 3. Optimal trajectories: analytical solution

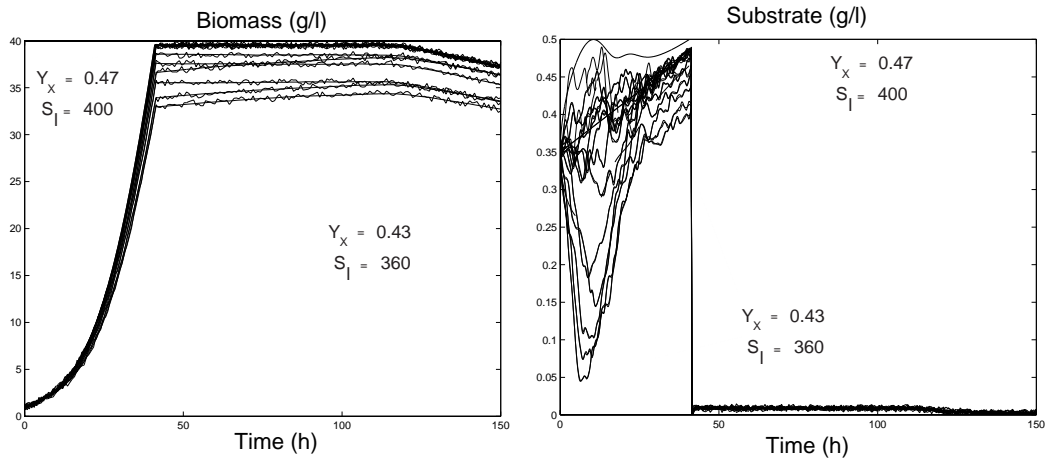


Fig. 4. Biomass and substrate profiles with back-offs for various parameter values and disturbance realizations

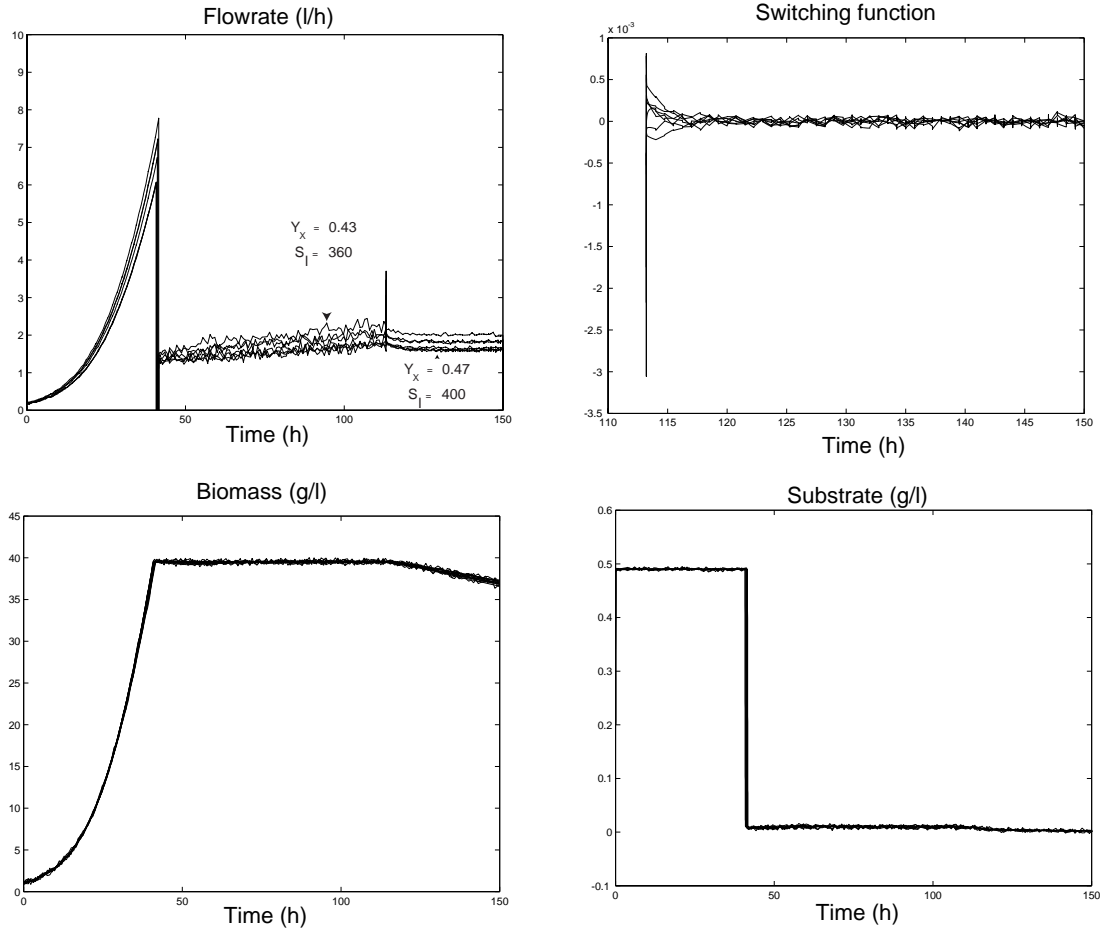


Fig. 5. Cascade optimization for various parameter values and disturbance realizations: no numerical reoptimization

α_m	$0.11 \frac{1}{h}$	Y_x	0.47
S_l	$400 \frac{g}{l}$	Y_p	1.2
K_l	0.006	M_x	$0.029 \frac{1}{h}$
θ_m	$0.004 \frac{1}{h}$	X_m	$40.0 \frac{g}{l}$
K_p	$0.0001 \frac{g}{l}$	S_m	$0.5 \frac{g}{l}$
K_i	$0.1 \frac{g}{l}$	K	$0.01 \frac{1}{h}$
t_f	150 h	u_m	$10 \frac{l}{h}$
X_o	$1 \frac{g}{l}$	P_o	$0 \frac{g}{l}$
S_o	$0.2 \frac{g}{l}$	V_o	250 l

Table 3: Model parameters, operating and initial conditions

t_1^s	0.019 h
t_2^s	40.897 h
t_3^s	40.997 h
t_4^s	113.467 h
t_5^s	149.998 h
Y_x	$0.43-0.47 \frac{1}{h}$
S_l	$360-400 \frac{g}{l}$

Table 4: Off-line switching times and range for parametric variations